

Workshop

Fakta

Kurs:	Systemutvecklare C/C++ Extended 2024
Klass:	SUVx24
Teknikområde:	Programspråket C, in- och utmatning, variabler, datatyper

Learning Target

Programmering i C och C++

Grundläggande begrepp inom programmering.

Skapa källkod med C syntax.

Grundläggande in- och utmatning

In- och utmatning av resultat från ett program med hjälp av standardfunktioner i C.

Användning av utvecklingsmiljöer

Använda *Visual Studio Code* (förkortat VSC) för att skapa källkod, kompilera och exekvera program skapat med programspråket C.

Introduktion

Att kunna göra beräkningar i datorprogram har varit en grundläggande del sedan programmering började. För att kunna utföra beräkningar krävs operatorer (**operators**) mellan värden (och/eller variabler). Operatorer delas in i grupper efter användningsområde, till exempel aritmetiska (**arithmetic**) för att använda de vanliga räknesätten eller logiska (**logical**) som kan utföra operationer med värden för sant och falskt. Tilldelningsoperatorer (**assignment**) kan användas för att både utföra en operation och samtidigt tilldela det nya värdet i en variabel. Jämförelseoperatorer (**comparison**) kommer väl till användning när vi ska utföra selektion, dvs olika val i programflödet.

Workshop

Förberedelse

Skapa en ny projektmapp (katalog) i den projektmapp du använder för kodning, typ:

```
C:\Chas\SUVx24\Workshop4\
```

Starta VSC på det sätt du tycker är lättast och öppna den nya projektmappen.

Innehåll

Uppgift 1

Att göra:

- Skapa programmet beskrivet nedan med C-kod i en ny fil kallad **workshop4-1.c**
- Koda, kompilera och exekvera och testa att det fungerar som tänkt.
- Rätta eventuella fel och provkör igen.

Skapa ett program som

- Räknar ut **area** och **omkrets** av en cirkel.
- Frågar efter radien eller diametern på cirkeln.
- Ta reda på rätt formel för beräkning av cirkelns area respektive omkrets.
 - Tänk på att ett tal i kvadrat är det samma som talet gånger sig självt en gång.
- Definiera π som en konstant med minst 4 decimalers noggrannhet
 - Hur många decimaler kan du använda för en float innan du behöver byta till en annan flyttalstyp?
- Gör beräkningarna med respektive formel.
- Mata ut resultaten för respektive mått med rätt enhet.

Workshop

Uppgift 2a

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop4-2a.c**

Skapa ett program som

- Frågar efter två heltal.
- Gör utskrifter som visar alla varianterna på jämförelse mellan två heltal.

Provkör med några olika heltalskombinationer, även negativa tal.

Uppgift 2b

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop4-2b.c**

Skapa ett program som

- Frågar efter två flyttal (du får själv välja mellan att använda float eller double).
- Gör utskrifter som visar alla varianterna på jämförelse mellan två flyttal.

Provkör med några olika flyttalskombinationer, även negativa tal och med olika många decimaler.

Uppgift 2c

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop4-2c.c**

Skapa ett program som

- Frågar efter två tecken.
- Gör utskrifter som visar alla varianterna på jämförelse mellan två tecken.

Provkör med några olika teckenkombinationer, även symboler typ @, !, %, etc).

Workshop

Uppgift 3

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop4-3.c**

Skapa ett program som

- Jämför uttryck med logiska operatorer.
- Skapa tre variabler med följande värden
 - a: 24, b: 15, c: 33
- Använd variablerna i följande uttryck
 - $a < b$ AND $a > c$
 - $a < b$ OR $a > c$
 - NOT ($a < b$ AND $a > c$)
 - NOT ($a < b$ OR $a > c$)
- Skriv ut de logiska uttrycken tillsammans med evalueringen, dvs slutresultatet av hela det logiska uttrycket).

Uppgift 4

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop3-4.c**

Skapa ett program som

- Visar alla varianter med tilldelningsoperatorer.
- Använd följande start på programmet och utöka med de andra varianterna:

```
// Declare variables
int a = 2;
int r = a;
printf("\nR is %i", r);

// Assignment operators
// Addition
r += a;
printf("\nR is %i", r);

// Subtraction
```

Workshop

Uppgift 5

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop4-5.c**

Skapa ett program som

- Utvärderar prioriteringen mellan de aritmetiska operatorerna.
- Skapa variabler med följande värden
 - a: 10, b: 20, c: 30, d: 40
- Använd variablerna i följande uttryck
 - $a + b / c + d$
 - $(a + b) / c + d$
 - $a + b / (c + d)$
 - $(a + b) / (c + d)$
- Gör en snygg utmatning av resultaten så att du förstår vad som händer med de olika beräkningarna betydelsen av 'operator precedence'.

Uppgift 6a

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop4-6a.c**

Skapa ett program som

- Konverterar mellan temperaturskalorna celsius och fahrenheit.
- Frågar efter ett värde i celsius.
- Googla formeln du behöver för att omvandla från celsius till fahrenheit.
- Gör beräkningen.
- Presenterar beräkningen med värdet i fahrenheit.

Workshop

Uppgift 6b

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop4-6b.c**

Skapa ett program som

- Konverterar mellan temperaturskalorna celsius och fahrenheit.
- Frågar efter ett värde i fahrenheit.
- Googla formeln du behöver för att omvandla från fahrenheit till celsius.
- Gör beräkningen.
- Presenterar beräkningen med värdet i celsius.

Uppgift 7

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop4-7.c**

Skapa ett program som

- Räknar ut **volym** av en cylinder.
- Frågar efter radien eller diametern på cirkeln.
- Ta reda på rätt formel för beräkning av cylinderns volym.
- Definiera PI som en konstant med minst 4 decimalers noggrannhet
- Gör beräkningarna med formeln.
- Mata ut resultaten för respektive mått med rätt enhet.